MULTIMEDIA UNIVERSITY®

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2019/2020

ECP1026 – ALGORITHMS AND DATA STRUCTURES

(TE / RE)

11 MARCH 2020
2.30 p.m. – 4.30 p.m.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 7 pages including cover page with 6 Questions only.
2. Attempt **FIVE** out of **SIX** questions. All questions carry equal marks and the distribution of marks for each question is given.
3. Please write all your answers in the Answer Booklet provided.

Question 1

(a) Consider a linked list shown in Figure Q1.1.

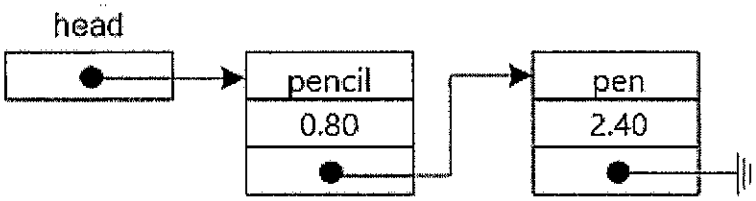

Figure Q1.1

The incomplete C program that implements the above list is given in Figure Q1.2.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//i. Define a self-referential structure type called Node
//   consisting of a 10-element array called item, a float
//   member called price, and a pointer member called next.

struct Node *initNode(char n[], float p);

int main() {
  struct Node *head;

  head = initNode("pencil", 0.8);

  //ii. Create a new node containing pen and its price, and
  //    have it linked to the node containing pencil

  while(head) {
    printf("%s (RM%.2f) -> ", head->item, head->price);
    head=head->next;
  }
  return 0;
}

struct Node *initNode(char n[], float p){
  struct Node *temp;

  //iii. Allocate memory for a new node dynamically &
  //     initialize the node with function parameters

  temp->next=NULL;
  return temp;
}
```

Figure Q1.2

Based on the comments given, complete the code. [4+3+5 marks]

Continued ...

Question 1 (cont.)

(b) Write the formatted output of the completed program in Question 1(a). [2 marks]

(c) Compare the implementation of a group of items using linked list and array of records in terms of memory allocation, access of data and ease in adding/removal of items. [6 marks]

Question 2

(a) Consider a binary tree shown in Figure Q2.1.

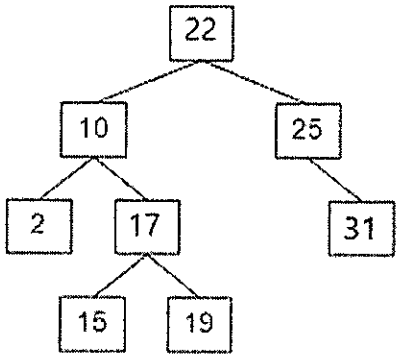

Figure Q2.1

Give the sequence of the visited nodes in preorder, postorder, and inorder traversals. [6 marks]

(b) Figure Q2.2 shows an incomplete C program that implements a binary tree using array index referencing method, traverse the tree in postorder, and print the contents of each node.

```
#include <stdio.h>
#define MAXNODES 9
struct Node {
    char data;
    int left;
    int right;
};
struct Node tree[MAXNODES] = {
    {'-', 1, 2},
    {'+', 3, 4},
    {'9',-1,-1},
    {'*', 5, 6},
    {'/', 7, 8},
    {'5',-1,-1},
    {'3',-1,-1},
    {'6',-1,-1},
    {'2',-1,-1}
};
```

Figure Q2.2

Continued...

Question 2 (cont.)

```
void visit(char data);
void postorder(int node);

int main(void){
  printf("\nPostorder traversal:\t"); postorder(0);
  return(0);
}

void visit(char data){ printf("%c ", data); }

// postorder function
```

Figure Q2.2 (cont.)

(i) Draw the tree that is being represented by the array of records. [9 marks]

(ii) Write the missing postorder function. [5 marks]

Question 3

(a) Consider the weighted digraph given in Figure Q3.1.

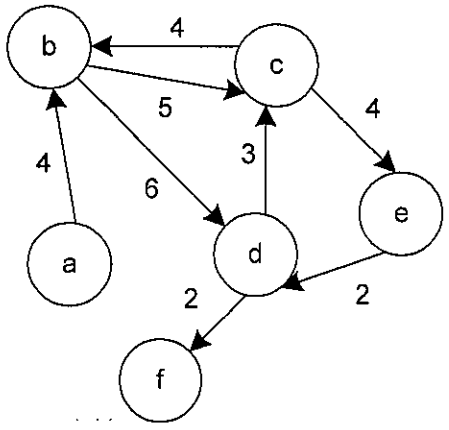

Figure Q3.1

(i) Represent the above graph in adjacency matrix form. [4 marks]

(ii) Write C declaration code that creates an array implementation of this graph. Use -1 to indicate no direct path from one node to another. [6 marks]

(b) Figure Q3.2 shows an incomplete implementation of set by using linear bitmap method. Based on the comments given, complete the program.

[4+2+4 marks]

Continued...

Question 3 (cont.)

```
#include <stdio.h>

typedef unsigned int Set;

int main() {
  int i;
  Set set1=0, set2=0, unionSet;

  //i. Initialize set1 = {4, 5, 6, 8} and set2 = {5, 8}

  //ii. Let unionSet store the union of set1 and set2

  for (i=0; i<16; i++){
  //iii. Print out the contents of unionSet
  }
  return 0;
}
```

Figure Q3.2

Question 4

(a) Explain the difference between direct recursion and mutual recursion. [2 marks]

(b) The Fibonacci sequence can be defined mathematically as follows:

$F_0=0$, $F_1=1$, $F_n=F_{n-1}+F_{n-2}$ where n=2,3,4,....

Each Fibonacci number is the sum of the two preceding numbers.

(i) Write the Fibonacci sequence if n is given as 7. [2 marks]

(ii) Implement the mathematical formulae to calculate Fibonacci value, F_n, for the given number, n, in ANSI-C using a direct recursive function. The function prototype is given as `int Fibonacci(int n);` [6 marks]

(iii) The program in Figure Q4.1 implements the mathematical formulae to calculate Fibonacci value, F_n, for the given number, n, in ANSI-C using iterative method. Calculate the worst time complexity of the program. Show the time complexity of each line (if any). [10 marks]

Continued ...

Question 4 (cont.)

```
#include <stdio.h>

int main()
{
  int n, first = 0, second = 1, next, c;
  printf("Enter the number of terms\n");
  scanf("%d", &n);
  printf("First %d terms of Fibonacci series are:\n", n);
  for (c = 0; c < n; c++)
  {
    if (c <= 1)
      next = c;
    else
    {
      next = first + second;
      first = second;
      second = next;
    }
    printf("%d\n", next);
  }
  return 0;
}
```

Figure Q4.1

Question 5

(a) Complete the C program in Figure Q5.1 of the implementation of linear search algorithm applied on an unsorted array. [8 marks]

```
#include <stdio.h>

// linear search function prototype

int main() {
  int a[] = {15, 27, 46, 23, 99, 32, 1, 8}, key, result;
  printf("Enter integer search key: ");
  scanf("%d", &key);

// linear search function call

  if (result != -1)
    printf("%d found in array element %d\n", key, result);
  else
    printf("%d not found\n", key);
  return 0;
}

// linear search function
```

Figure Q5.1

Continued...

Question 5 (cont.)

(b) Consider the five keys below:

89, 18, 49, 58, 9

Using the hash function: index = key % size, determine the index of each keys. Then, show how the keys would go into a hash table of size 10 after each insertion by using the following methods in collision resolution. [2+5+5 marks]

(i) Linear probing
(ii) Quadratic probing

Question 6

(a) Explain the bubblesort algorithm. Why bubblesort algorithm is considered not practical compared to selection sort or insertion sort? [4+2 marks]

(b) Complete the insertion sort function that arranges a given string in alphabetical order in Figure Q6.1 based on the comments given. [1+3+1 marks]

```
void insertionSort(char *a, int n){
  char curr;
  int j,k;

  for(j=0;j<n;j++){
    //i. store current element

    //ii. start from last element in the sorted section;
    //while element to be inserted is smaller than sorted element,
    //move the sorted element back one position

    //iii. insert element in correct position in already sorted
    //portion of array

    printf("%2d: %s\n",j,a);
  }
}
```

Figure Q6.1

(a) Sort the string **937284615** in increasing order using insertion sort. Show the result at each step. [9 marks]

End of Paper